

Pohledy

Pohledy

- Pohledy se dají chápat jako náhled nad databázovými objekty – tabulkami, jinými pohledy. Jedná se v podstatě o uložený dotaz (SELECT).
- S pohledem se pracuje stejně jako s tabulkou.
- Vnořený dotaz s aliasem se nazývá inline pohled.

Př.:

```
SELECT jmeno, prijmeni, hodnost
FROM zamestnanec z,
    (SELECT zamestnani_id, nazev_pozice AS
        hodnost FROM zamestnani
    ) zi
WHERE z.zamestnani_id = zi.zamestnani_id;
```

Vlastnosti pohledů

- Poskytují dodatečné zabezpečení tabulky – omezení přístupu k nepovoleným, citlivým atributům nebo k řádkům tabulky (např. atributy heslo, plat mohou být v pohledu vynechány, přístupová práva k originální tabulce omezena).
- Zjednodušují výběr dat z komplexních struktur – navenek se pohled tváří jako jedna tabulka, ale fyzicky se data mohou spojovat přes několik různých tabulek různým způsobem (např. pro zjednodušení tvorby dotazů v aplikacích při zachování flexibility/strukturovanosti návrhu databáze).
- Mohou zobrazovat data různým způsobem oproti zdrojové tabulce. Data ve zdrojové tabulce se neovlivní. Např.: přejmenování atributů, spojení atributů.
- Možnost odstínění vlivu změny zdrojové tabulky v aplikaci – poskytují logickou vrstvu mezi aplikací a zdrojovými tabulkami.

Stinná stránka pohledů

- Pohledy mají i svou stinnou stránku – vzhledem k tomu, že se s nimi pracuje jako s tabulkou, tak bez Explain plan ani nemusíme tušit, že vlastně pracujeme s pohledem a nikoli s tabulkou.

Aktualizace dat pohledu I.

- Data ve zdrojových tabulkách lze prostřednictvím pohledu upravovat (I/U/D) při splnění určitých podmínek.
- Pohled nesmí obsahovat:
 - Množinové operace (UNION),
 - operátor DISTINCT,
 - agregační nebo analytické funkce,
 - poddotaz v seznamu sloupců, které se vybírají dotazem,
 - klauzule CONNECT BY, START WITH,
 - spojení více tabulek, kromě výjimek:
 - Úpravy se musí týkat pouze jedné tabulky,
 - vložení záznamu je možné pouze bez uvedené klauzule WITH CHECK OPTION (pro úpravu a smazání záznamu klauzule uvedena být může).

Aktualizace dat pohledu II.

- Můžete zakázat změnu dat ve zdrojových tabulkách prostřednictvím pohledu tím, že ho nastavíte READ ONLY:

```
CREATE VIEW RO_ZAM  
AS SELECT ZAMESTNANEC_ID, JMENO, PRIJMENI  
FROM ZAMESTNANEC  
WITH READ ONLY;
```

Vytvoření pohledu

```
CREATE [OR REPLACE] [NOFORCE | FORCE] VIEW <název pohledu>  
AS <poddotaz>  
[WITH READ ONLY |  
  CHECK OPTION [CONSTRAINT <název omezení>]];
```

- Vytvoří pohled <název pohledu>.
- Pokud je uvedeno **FORCE**, databázové objekty (tabulky, pohledy) uvedené v <poddotazu> nemusí existovat. Výchozí hodnota je **NOFORCE**.
- Klauzule **READ ONLY** určuje, že pohled bude pouze pro zobrazení dat. Nebude možné vkládat, editovat, mazat záznamy ze zdrojových tabulek prostřednictvím pohledu.
- **CHECK OPTION** – upravovat (I/U/D) lze pouze sloupce a řádky, které view vrátí. **CONSTRAINT** <název omezení> definuje jméno tohoto omezení, jinak se použije automatické jméno (**SYS_Cn** – n je integer).
- Smazání pohledu: **DROP VIEW** <název pohledu>;

Vytvoření pohledu – příklad

```
CREATE NOFORCE VIEW uklizecky AS  
  SELECT jmeno, prijmeni,  
    zamestnani_id AS pozice  
FROM zamestnanec  
WHERE zamestnani_id = (  
    SELECT zamestnani_id FROM zamestnani  
    WHERE nazev_pozice = 'Uklízečka')  
WITH CHECK OPTION CONSTRAINT jen_uklizecky;
```

- Vytvoří pohled nad zdrojovými tabulkami zamestnanec, zamestnani. Tabulky v době tvorby pohledu musí existovat. Prostřednictvím pohledu lze měnit pouze zaměstnance, kteří mají pozici 'Uklízečka' – vkládat, editovat i mazat.

Materializovaný pohled

Materializovaný pohled

- Liší se od normálního pohledu tím, že výsledek dotazu je uložen (materializován) jako samostatný databázový objekt (tabulka) – dochází zde k replikaci dat.
- Při odstranění zdrojových tabulek zůstanou materializované pohledy zachovány.
- Vytvoření Materializovaného pohledu:

CREATE MATERIALIZED VIEW <název> AS <poddotaz>;

- Když v SQL dotazu, kterým vytvoříte Materializovaný pohled použijete ORDER BY klauzuli, tak budou záznamy v pohledu seříděny pouze při jeho vytvoření. Později při jeho aktualizaci se toto pořadí neudrží (tato informace není součástí uloženého materializovaného pohledu).

Aktualizace dat I.

- Když změníte data zdrojové tabulky, materializovaný pohled standardně neprovede žádnou synchronizaci a bude obsahovat neaktuální data. K tomu je nutné specifikovat jak a kdy se provede aktualizace dat:
- Jak se provede aktualizace dat:
 - REFRESH COMPLETE: Celý obsah Materializovaného pohledu se smaže a znovu se vytvoří z dat zdrojových tabulek.
 - REFRESH FAST: Dojde k pouhé aktualizaci dat Materializovaného pohledu.
 - REFRESH FORCE: Pokusí se provést REFRESH FAST a když se tato operace nezdaří, provede REFRESH COMPLETE. Výchozí metoda.
 - NEVER REFRESH: Zakáže změny v Materializovaném pohledu.

Refresh Complete vs Fast

- REFRESH COMPLETE:
 - Ideální, když se ve zdrojových tabulkách změnilo hodně dat.
- REFRESH FAST:
 - Pouze v Oracle Enterprise Edition.
 - Ideální, když se ve zdrojových tabulkách změnilo málo dat.
 - Vyžaduje další strukturu – Materialized View Log, která obsahuje incrementální změny ve zdrojové tabulce.

Materialized View Log

- Fast Refresh Materializovaného pohledu vyžaduje nejen Oracle Enterprise Edition, ale také, aby na zdrojových tabulkách byl nastaven log, ve kterém budou informace o incrementálních změnách těchto tabulek.

- Jak takový log vytvořit?

```
create materialized view log on employees with rowid;
```

- Pozor! Udržování této struktury je relativně náročné.

Aktualizace dat II.

- Kdy se provede aktualizace dat:
 - ON DEMAND: Data Materializovaného pohledu jsou aktualizována buď manuálním způsobem, nebo periodicky. Mezi dvěma aktualizacemi může Materializovaný pohled obsahovat neaktuální data.
 - ON COMMIT: Data jsou aktualizována automaticky ve stejné transakci, která modifikuje zdrojové tabulky. Materializovaný pohled vždy obsahuje aktuální data.
- Typy aktualizace dat je možné nastavit při vytváření Materializovaného pohledu, nebo při jeho změně:

```
ALTER MATERIALIZED VIEW <název>
```

```
REFRESH FORCE ON DEMAND;
```

Práva pro práci s MV

- Spustíte jako administrátor:

GRANT CREATE ANY MATERIALIZED VIEW,

ALTER ANY MATERIALIZED VIEW,

DROP ANY MATERIALIZED VIEW,

QUERY REWRITE,

GLOBAL QUERY REWRITE

TO hr ← Schéma, kterému přidělujeme oprávnění

WITH ADMIN OPTION;

Ruční aktualizace

- Materializovaný pohled:

```
create materialized view hr.emps_mv as  
(select * from hr.employees where salary > 10000);
```

- Po změně zdrojových dat zůstane beze změny:

```
update employees set salary = 10000 where email = 'SKING';
```

- Ruční aktualizaci provedete takto:

```
execute dbms_mview.refresh(list => 'emps_mv');
```

- Můžete také specifikovat typ aktualizace ('c' = complete, 'f' = fast, '?' = force):

```
execute dbms_mview.refresh(list => 'emps_mv', method => '?');
```

- V balíčku dbms_mview jsou také procedury pro aktualizaci všech MV a aktualizaci pohledů, které jsou závislé na vybraných tabulkách.

Periodická aktualizace

- Můžete také nastavit periodickou aktualizaci MV:

```
alter materialized view emps_mv
```

```
refresh complete on demand
```

```
start with sysdate
```

```
next sysdate+to_dsinterval('0 00:01:00');
```

Spustí refresh
každou minutu.

Funkce, která zkonvertuje text do datového typu
INTERVAL DAY TO SECOND

- Tímto zkontrolujete běh jobu:

```
select * from user_jobs;
```

